Funciones

1) Diseñar una función que calcule el promedio de varios números introducidos por el teclado.
2) Diseñar una función que calcule la potencia enésima de un número, es decir que calcule X n para X, real y n entero
3) Escribir una función "EsMultiplo" que sirva para determinar si un número es múltiplo de otra.
4) Utilizando como primitivas las operaciones *, +, - y /, escriba algoritmos que implementen las siguientes primitivas:
   - EsDivisiblePor: recibe dos números enteros N y D, y devuelve verdadero si N es divisible por D, y falso en caso contrario.
   - ValorAbsoluto: recibe un número entero N, y devuelve el valor absoluto de N.
   - RaizCuadradaEntera: recibe un número natural N, y devuelve la raíz entera (sin decimales) de N. Ej: si N es 10, se devuelve 3.
   - EsCuadradoPerfecto: recibe un número natural N, y devuelve verdadero si N es cuadrado perfecto; falso en caso contrario.
   - Resto: recibe dos nros enteros N y D, y devuelve el resto de dividir N por D.
   - Potencia: recibe una base real B y un exponente entero E, y devuelve B elevado a la E.
   - CantidadDeCifras: recibe un número entero N, y devuelve la cantidad de dígitos de N (ej: si N = 3421, devuelve 4)

   Obs.: el operador “/" representa la división. Cuando se lo aplica a dos enteros Z1 y Z2, Z1 / Z2 corresponde a la división entera entre Z1 y Z2. Cuando se los aplica a reales R1 y R2, la división R1 / R2 corresponde a la división real.
5) Escriba un algoritmo que reciba como entrada un número natural entre 1 y 999, y muestre como salida el número romano correspondiente. Ejemplo: si se entra 125, deberá mostrarse cxxv; si se entra 40, deberá mostrarse xl. Pista: piense cuál es el criterio que se usa para armar los números romanos. Los símbolos I, V, X, L, C, D, M corresponden a 1, 5, 10, 50, 100, 500 y 1000, respectivamente.
6) El Tercer Teorema de Fermat enuncia que no existen números enteros x, y, z, tales que para un número natural n, n > 2, se verifique que xn+yn=zn. Sin embargo, el matemático francés Pierre Fermat (1601-1665) tan sólo enunció el teorema anterior, pero no pudo demostrarlo. . . (según él, porque la hoja que tenía en ese momento no le alcanzaba). Quizás el teorema sea falso. Para asegurar eso habría que encontrar un contraejemplo, esto es, bastaría encontrar tres números x; y; z y un valor n mayor que 2 tales que xn+yn=zn.

   Obs.: para n = 2, lo anterior puede satisfacerse; Ejemplo: 32 +42 = 52).

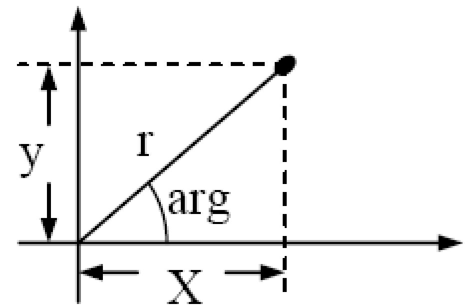
Escriba un algoritmo que determine si el Tercer Teorema de Fermat se cumple para cualquier entero entre 0 y 20, con n = 3 y n = 4. (Es decir, el algoritmo deberá devolver verdadero si no existen valores x; y; z comprendidos entre 0 y 20, tales que para n=3 y n=4, resulte xn+yn=zn).

7) La conjetura de Goldbach expresa que todo número natural par puede expresarse como la suma de dos naturales primos. Ejemplo: 18=11+7, 20=17+3, 30=11+19. Por ser una “conjetura" y no un “teorema", el enunciado anterior podría llegar a ser falso. Es decir, nadie ha demostrado que lo anterior siempre se cumple, pero nadie tampoco ha encontrado un contraejemplo. Escriba un algoritmo que devuelva verdadero en caso de que la conjetura de Goldbach se verifique para todos los números pares entre 100 y 1000, y falso en caso contrario.

8) Escribir un programa que calcule el número de billetes de 10.000, 5.000, 1,000, así como de monedas de 500, 100, 25, 5 y 1 pesetas para desglosar una cantidad, C, de pesetas (menor de 2.147.483.647), de forma que se necesite la menor cantidad de monedas y billetes de cada tipo.

9) Leer desde el teclado las coordenadas (x,y) de un punto y, mediante un menú, elegir una entre las siguientes opciones:

- Comprobar si el punto pertenece a una circunferencia de radio 10 y centro (0,0)
- Averiguar el cuadrante en el que se encuentra el punto.
- Pasar las coordenadas cartesianas (x,y) a polares (r,arg).



10) Escribir un programa que calcule la sumatoria:

$$\sum_{i=1}^{S} (-1)^i * \frac{1}{i^2}$$

Donde S es un número entero positivo introducido por teclado.
Solución: El límite de esa expresión cuando S tiende a infinito es: -0.822467.

11) Escribir un programa que permita convertir grados Fahrenheit a Celsius y grados Celsius a Fahrenheit.
El programa presentará el siguiente menú:
1) Conversión de Celsius a Fahrenheit
2) Conversión de Fahrenheit a Celsius
3) Salir del programa.

Nota: Cada conversión se efectuará por medio de funciones, una que convertirá de grados Celsius a grados Fahrenheit y otra que haga justo lo contrario.

12) Se desea realizar un programa que permita hacer cálculos con polinomios de segundo grado.
El programa deberá presentar un menú en pantalla que facilite las siguientes opciones:
1. Leer un polinomio
2. Escribir un polinomio en su forma habitual
3. Evaluar un polinomio en un punto
4. Calcular el polinomio derivado
0. Salir
Se deberán utilizar los siguientes subprogramas:
LEER mediante este subprograma se introducen los números enteros que representan los coeficientes del polinomio.
ESCRIBIR es un subprograma que permite la escritura de un polinomio en la forma: Ax^2+Bx+C
Ejemplos válidos: 5x^2+3x+5 5x^2-3x+5 5x^2-5
Ejemplos no válidos: 5x^2+ -3x+5 5x^2-0x+5
EVALUAR es un subprograma que retorna el valor de un polinomio para un número real que es introducido como parámetro.
DERIVAR es un subprograma que calcula el polinomio derivada de uno dado.
Nota: No está permitido el uso de variables globales.

13) Escribir un programa que realice la descomposición en factores primos de un número introducido por teclado.
El programa deberá ir escribiendo la tabla de los factores primos, a medida que los va calculando, tal como muestra el ejemplo siguiente:

Introduce un Nº entero -> **84**

Nº	Factores primos
84	2
42	2
21	3
7	7
1	

Ayuda: Se deberá utilizar una función que nos diga si un número es primo o no.

14) Crear una función que intercambie el contenido de dos variables. Para ello se pasarán como parámetros las direcciones de las variables.
Para probar la función escribir un programa que pida los datos por pantalla y muestre los contenidos después de llamar a la función.

Funciones Recursivas

15) Escribir una función que calcule el factorial de un número y utilizar ésta en un programa que muestre el siguiente menú.
1. Factorial de un número
2. Cálculo de e
3. Cálculo de e^x
0. Salir
Nota 1: El cálculo de *e* debe hacerse con la siguiente expresión matemática:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \ldots$$

Nota 2: *ex* puede calcularse mediante la fórmula:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$$

Nota 3: La precisión con la que se obtiene el resultado (*e* o *e x*) depende del último valor añadido en la correspondiente serie.

16) El desarrollo en serie de Taylor de la función coseno es:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots$$

donde el ángulo x viene expresado en radianes.
Escribir un programa que calcule el valor aproximado de coseno (x), utilizando para ello los N primeros términos de la serie de Taylor. El número de términos N de la serie dependerá del valor del último, es decir, se añadirán términos a la serie hasta que el valor absoluto del último término añadido sea menor que 0.0005.

Strings

17) Escribir un programa que pida una frase acabada en un punto y cuente las palabras que contiene.
18) Escribir un programa que pida una palabra y cuente el número de vocales y consonantes que contiene.
19) Escribir un programa que lea una frase y la escriba en mayúsculas.
20) Escribir un programa que sirva para generar códigos de usuario por el procedimiento siguiente: Tiene que leer los nombres y el apellido de una persona y devolver un código de usuario formado por las tres primeras letras del, las tres primeras letras del segundo nombre y las tres primeras letras del primer nombre. Por ejemplo, si la cadena de entrada es: “JORGE GUIDO SANTOS” debe devolver “SANGUIJOR”.
21) Diseñar una función "aMayusculas()" que convierta una cadena de texto en mayúsculas

22) Escriba un algoritmo que muestre todas las palabras de cuatro letras que se pueden armar con las letras c, e, p, y a. Obs.: se asume que una “palabra" es una secuencia cualquiera de las cuatro letras dadas, posiblemente repetidas. Ej: ecpa, ppca y eeee son algunas de las palabras que deben mostrarse.

23) Escribir un programa que cuente de un texto introducido por teclado:
- N.º de caracteres en blanco
- N.º de dígitos
- N.º de letras
- N.º de líneas
- N.º de otros caracteres

Nota 1: Se deben crear sendas funciones para comprobar si un carácter es numérico o alfanumérico.

Nota 2: La función *getchar()* permite leer un carácter de teclado.

24) Realizar un programa que lea una cadena de caracteres de una longitud menor de 80 y visualice los caracteres de la siguiente forma: primero, último, segundo, penúltimo, tercero, antepenúltimo, ...

25) Escribir una función que cambie las letras mayúsculas de una cadena a minúsculas y viceversa. El programa principal pedirá una cadena por pantalla y se la pasará a dicha función esperando el resultado correcto que se mostrará por pantalla.

26) Escribir un programa que pida primero un carácter por teclado y que luego pida una cadena. El programa calculará cuántos caracteres tiene la cadena hasta que lea el carácter introducido primero. Se deberá mostrar un mensaje en pantalla con el número de caracteres introducidos hasta llegar al carácter primero.

27) Realizar un programa que lea una cadena de caracteres con espacios en blanco excesivos: elimine los espacios en blanco iniciales y finales y sólo deje uno entre cada dos palabras.

28) Escribir un programa que pida dos cadenas (de longitud máxima 10 caracteres) y muestre por pantalla el resultado de las siguientes operaciones:
- Obtener la longitud de ambas cadenas
- Comparar alfabéticamente ambas cadenas indicando si son iguales o bien cuál es la mayor y cuál la menor.
- Concatenar la segunda cadena al final de la primera, dejando un espacio blanco entre ambas.
- Copiar el contenido de la segunda cadena en la primera.

Estructuras

29) Escribir un programa que lea dos números complejos y permita realizar con ellos las siguientes operaciones aritméticas: suma, resta, multiplicación y división

Nota 1: Se debe crear una función de permita leer un número complejo (su parte real y su parte imaginaria).
Nota 2: Se debe crear una función de permita pasar un número complejo en forma parte real y parte imaginaria a módulo y argumento. Se debe crear una función que permita pasar un número complejo en forma módulo y argumento a parte real y parte imaginaria.
Nota 3: La suma y resta de números complejos se obtiene sumando, o restando, las partes reales y las partes complejas.
El producto de dos números complejos se obtiene multiplicando sus módulos y sumando sus argumentos.
El cociente de dos números complejos se obtiene dividiendo sus módulos y restando sus argumentos.

30) Crear un programa que permita introducir cierta información relativa a los vuelos diarios que parten de un aeropuerto en un array formado por registros. Cada registro contendrá la siguiente información sobre el vuelo correspondiente:
   a) Número de vuelo (No tiene por qué coincidir con el índice del array)
   b) Hora de partida (En dos campos):
      Hora: 0..23
      Minutos: 0..59
   c) Origen del vuelo: Cadena de caracteres
   d) Destino del vuelo: Cadena de caracteres
   e) Número de pasajeros: Entero

   Una vez introducidos los datos de todos los vuelos se preguntará si se desea obtener información de algún vuelo. En caso de que el usuario responda afirmativamente se pedirá el número de vuelo. El programa buscará el vuelo en al array y accederá a la información que contiene a partir de su número de vuelo, mostrando por pantalla todos sus datos.
   El programa se ejecutará repetitivamente hasta que el usuario indique que no desea obtener más información de ningún vuelo.
   Nota: Se deben diseñar las funciones que visualicen un vuelo, busquen un vuelo en el array, introduzcan la información de un vuelo en el array, etc...

Archivos

31) Programa que implemente una agenda.
   Se guardarán los siguientes datos por persona:
   - *Nombre:* Máximo 15 caracteres.
   - *Apellidos:* Máximo 35 caracteres.
   - *Sobrenombre:* Máximo 10 caracteres.
   - *Teléfono:* Máximo 10 caracteres.
   - *Fecha de nacimiento:* 8 caracteres (con el formato dd/mm/aa, pudiendo ser espacios los dígitos del año

   Se creará un menú con las siguientes opciones:

1. *Alta* de una nueva persona a la agenda con los correspondientes datos. 2. *Eliminar* a una persona de la agenda.
3. *Búsqueda* de un nombre en particular (La búsqueda se hará por sobrenombre)
4. *Listado* de todas las personas empezando por la primera introducida (Para pasar a la siguiente se deberá pulsar intro).
5. *Guardar* agenda en disco.
0. *Salir*

32) Se deberá escribir un programa que permita introducir y consultar la tabla periódica de los elementos químicos. Para ello, se construirá un menú con las siguientes opciones:
    1) Introducir elementos de la tabla periódica
    2) Listar todos los elementos de tabla periódica
    3) Mostrar elemento de la tabla periódica por número atómico
    4) Mostrar elemento de la tabla periódica por símbolo
    5) Salir

    La **opción 1** preguntará cuántos elementos de la tabla periódica se quieren introducir e irá preguntando sucesivamente por ellos. Evidentemente, no es necesario rellenar el array con todos los elementos de la tabla periódica.
    La **opción 2** listará la información acerca de todos los elementos químicos introducidos ordenados por número atómico.
    La **opción 3** pedirá el número atómico del elemento que se quiere consultar y mostrará por pantalla la información correspondiente a dicho elemento.
    La **opción 4** pedirá el símbolo del elemento que se quiere consultar y mostrará por pantalla la información correspondiente a dicho elemento.
    Se guardará la siguiente información sobre cada elemento.
    1) Símbolo del elemento
    2) Nombre completo
    3) Peso Atómico

    El número atómico vendrá indicado por la posición en el array de elementos que representa la tabla periódica, y por tanto no será necesario almacenar esta información en el registro correspondiente.
33) Crear un programa que abra un fichero y escriba números enteros y otro programa calcule el valor máximo (si hay varios basta con uno de ello) y la media de todos los números contenidos en el fichero anterior:
    Ayuda: Utilícense las funciones *fprintf* y *fscanf*.
34) Leer completamente un fichero de texto, carácter a carácter (o en cantidades mayores, para que sea más rápido). El programa debe contar las vocales, los caracteres alfabéticos y los dígitos que hay en el fichero.
35) Crear un programa que abra un fichero y escriba en él dos cadenas, cada una acabada con el carácter de nueva línea y otro

programa que lea la segunda cadena escrita en el mismo fichero.
Comprobar que el fichero existe y visualizar su contenido con un editor de textos (p.e. el *Block de Notas* de Windows)
Nota: Escribir una cadena con espacios intercalados.
Ayuda: Utilícense las funciones *fputs* y *fgets*. Consultar la ayuda.

36) Escribir un programa que tome caracteres de teclado y, de uno en uno, los escriba en un fichero cuyo nombre es previamente pedido por pantalla.

37) Escribir un programa que use dos ficheros: uno de lectura y otro de escritura. El programa leerá los caracteres de un fichero, y tras una operación de cambio de mayúsculas a minúsculas y viceversa, escogida por el usuario, los escribirá en un segundo fichero.

38) Escribir el código necesario en el programa de la agenda para que se añadan cuatro opciones más. De esta forma, el menú quedará de la siguiente manera:
1. Alta de una nueva persona a la agenda con los correspondientes datos.
2. Eliminar a una persona de la agenda.
3. Búsqueda de un nombre en particular (La búsqueda se hará por sobrenombre)
4. Listado de todas las personas empezando por la primera introducida (Para pasar a la siguiente se deberá pulsar intro).
5. Leer la agenda desde el disco (formato binario). El nombre del fichero será *agenda.bin*.
6. Guardar la agenda en el disco (formato binario). El nombre del fichero será *agenda.bin*.
7. Guardar la agenda en el disco para imprimir (en modo texto). El nombre del fichero será *agenda.txt*.
8. Guardar en un fichero de texto información de las personas que cumplen años en un determinado mes (cuyo número se leerá desde teclado). El nombre del fichero será el número del mes seguido de la extensión txt.
Se guardará la fecha de nacimiento, el sobrenombre y el teléfono, ocupando la información de cada persona una línea. La información deberá estar encolumnada debajo de una cabecera, tal como se muestra en el ejemplo:

```
FECHA DE NACIMIENTO     SOBRENOMBRE          TELEFONO
13/12/85                Koldo                946478383
09/12/90                Ana                  653765432
```

0. Salir

39) Se necesita construir un programa que a partir de la fórmula de un compuesto químico (supuestamente puro) y su peso en gramos obtenga la cantidad que dicho compuesto contiene de cada uno de los elementos químicos que lo forman.
Para ello se escribirán dos programas:

El **primero** de ellos escribirá en disco la tabla periódica. Por cada elemento de la tabla periódica se guardará la siguiente información:

- Símbolo del elemento
- Nombre completo
- Peso Atómico

El número atómico de cada elemento vendrá representado por su posición en la tabla periódica.

El **segundo** de ellos leerá el fichero con la tabla periódica introducido con el programa anterior y presentará un menú con las siguientes opciones.

1) Listar tabla periódica
2) Mostrar elemento de la tabla periódica
3) Obtener la composición de un compuesto químico
4) Salir

La **opción 1** mostrará un listado con todos los elementos de la tabla periódica, parando la ejecución cuando se llene una pantalla para permitir verlos todos.

La **opción 2** pedirá el símbolo de un elemento químico, lo buscará en la tabla periódica y mostrará toda la información relativa. Es decir, su número atómico, el símbolo del mismo, su nombre completo y su peso atómico.

La **opción 3** permitirá introducir la fórmula del compuesto químico. Para ello irá preguntará primero por el número de átomos constitutivos y luego por el nombre y número de átomos del elemento en el compuesto. Posteriormente, se pedirá el peso en gramos del compuesto a analizar.

Una vez calculada con la tabla periódica, a partir de las proporciones obtenidas de la fórmula, la cantidad en gramos de cada uno de los elementos se mostrará el informe con la composición en gramos del compuesto.

Finalmente, la **opción 4** permite salir del programa.

40) Crear un archivo binario llamado BIN.DAT el cuál contendrá estructuras con los siguientes campos:

    char nombre[10]
    int edad

    Terminar el ingreso de datos cuando se presione la tecla ESC. Una vez terminado el ingreso de los datos, abrir el archivo con un editor de texto y analizar su contenido.

41) Abrir el archivo BIN.DAT para lectura y mostrar su contenido en pantalla.
42) Abrir el archivo BIN.DAT para agregar datos, una vez finalizado el ingreso de datos, mostrar en pantalla el contenido completo del archivo. En el caso que aparezca el modificador /b en la línea de comandos se debe generar un backup del archivo usando una función.
43) Realizar un programa que me permita modificar los datos de uno o varios registros del archivo BIN.DAT. Para ello se debe ingresar el apellido del registro que se desea modificar.

44) Para el sector de ventas de una empresa se necesita realizar un programa con las siguientes características

En el archivo producto.dat se deben guardar todos los productos que existen para la venta y los datos que se ingresan para cada producto son los siguientes:

Código de artículo	6 caracteres de longitud y el primer carácter debe ser X
Descripción	Puede tener un máximo de 20 caracteres
Precio	Precio del producto
Cantidad	Cantidad en existencia del producto

En el archivo ventas.dat se deben guardar las ventas realizadas y los datos que se guardan en este archivo son:

Código de artículo
Cantidad vendida
Fecha de venta

El programa debe contar con un menú de opciones que tenga Carga de productos, Venta de productos y Salir del programa

Carga de productos: Se ingresan los datos relativos a cada producto y se guarda en el archivo producto.dat, si el código ya existe en el archivo se debe informar al usuario y no se debe guardar y si el código no existe se debe agregar al final. Todos los datos ingresados deben ser validados por funciones.

Venta de productos: Se ingresa el código de producto que se desea vender y la cantidad a vender. Se busca el código en el archivo producto.dat y si la cantidad existente es suficiente se realiza la venta escribiendo en el archivo ventas.dat el código, la cantidad vendida y la fecha. Por otra parte se debe actualizar la cantidad existente en el archivo producto.dat.

Si la cantidad en existencia es insuficiente se debe mostrar un mensaje de error y no se realizará la venta.